

Dirichlet Fragmentation Processes: A Useful Variant of Fragmentation Processes for Modelling Hierarchical Data

Hong Ge
University of Cambridge
hg344@cam.ac.uk

Yarin Gal
University of Cambridge
yg279@cam.ac.uk

Zoubin Ghahramani
University of Cambridge
zg201@cam.ac.uk

September 17, 2015

Abstract

Tree structures are ubiquitous in data across many domains, and many datasets are naturally modelled by unobserved tree structures. In this paper, first we review the theory of random fragmentation processes [Bertoin, 2006], and a number of existing methods for modelling trees, including the popular nested Chinese restaurant process (nCRP). Then we define a general class of probability distributions over trees: the *Dirichlet fragmentation process* (DFP) through a novel combination of the theory of Dirichlet processes and random fragmentation processes. This DFP presents a stick-breaking construction, and relates to the nCRP in the same way the Dirichlet process relates to the Chinese restaurant process. Furthermore, we develop a novel hierarchical mixture model with the DFP, and empirically compare the new model to similar models in machine learning. Experiments show the DFP mixture model to be convincingly better than existing state-of-the-art approaches for hierarchical clustering and density modelling.

The process of random fragmentation is common to many areas, such as the degradation of large polymer chains in chemistry, or the evolution of phylogenetic trees in biology. An elegant mathematical tool for describing such phenomena is the *fragmentation process* (FP) [Bertoin, 2006]. As a concrete example of a FP, consider a stick of unit length. At every time point, the stick breaks into two smaller pieces. Then, each of the resulting smaller sticks independently repeats the procedure, and the process continues ad infinitum. This process can be described with the FP framework, and generalised to arbitrary distributions over the splits of the stick, breaking times, and number of splits.

The process of fragmentation can be interpreted as inducing a tree structure. In the probability theory community, Aldous [1991] has worked on binary fragmentation trees and used a symmetric beta distribution as the fragmentation operator for binary trees. McCullagh et al. [2008] has worked on the theoretical aspect of Bertoin [2006]’s relation



Figure 1: Recursive Stick Breaking. The plot on the left shows an example of recursive breaking; At the first level, the unit-size stick breaks into infinitely many sub-sticks. The first 3 sticks are illustrated and the remaining sticks are represented by dots. Then at the second level, a similar stick breaking process is applied to each sub-stick. This recursive stick-breaking process repeats until a pre-determined maximum depth is reached. The plot in the middle shows the resulting tree structure by discarding stick sizes. The plot on the right shows the sequence indexing scheme.

to tree priors, studying both binary and multifurcating trees. Teh et al. [2011] have recently began studying the relation between fragmentation and coagulation processes, and relating these to practical applications in machine learning. Apart from the last work, the literature has mostly concentrated on theoretical aspects of the FP, and pragmatic aspects of the process have been largely overlooked.

The rest of this paper is organised as follows. Section 1 briefly reviews the result of fragmentation processes (FP) as introduced in Bertoin [2006], and the nested Chinese restaurant process [Blei et al., 2010]. This lays the way for a general probabilistic framework for modelling trees. In Section 2, we derive a useful variant of fragmentation processes – the *Dirichlet fragmentation process* (DFP) – through a combination of the theory of Dirichlet processes and fragmentation process. A notable property of the DFP is that it relates to the nCRP in the same way the Dirichlet process relates to the Chinese restaurant process, that is the DFP forms the underlying de Finetti measure of the nCRP. Inspired by this property, in Section 3 we develop a hierarchical infinite mixture model with the DFP prior as its mixing distribution, in the same spirit as using the Dirichlet process prior as the mixing distribution for an infinite mixture model. Furthermore, in Section 4 we describe an associated effective yet simple sampling procedure for the DFP mixture model. Finally, in Section 5 we assess the model with a set of experiments for density estimation and hierarchical clustering, demonstrating an improvement on existing state-of-the-art approaches.

1 Preliminaries

We begin by briefly reviewing the fragmentation process and nested Chinese restaurant process upon which our new model is based. The relation between the two will become clear in the next sections.

Throughout this paper, we will use finite-length sequences of natural numbers as our index set on the nodes in a tree, i.e. we let $\omega = (\omega_1, \omega_2, \dots, \omega_L)$ denote a length- L sequence of positive integers, $\omega_i \in \mathbb{N}$. We denote the zero-length string as $\omega = \Delta$ and use $|\omega|$ to indicate the length of sequence ω . When viewing these strings as node indices in a tree, $(\omega\omega_i: \omega_i \in \mathbb{N})$ are the children of ω , and $\Delta \preceq \omega' \prec \omega$ are the ancestors of ω , and Δ is the root of the tree.

1.1 Fragmentation Processes

To give a more concrete description fragmentation processes, first recall the stick-breaking example of fragmentation processes. We use $\boldsymbol{\pi}(t)$ to denote the set of sub-sticks present at each time $t \in \mathbb{R}^+$ (the set of non-negative real numbers), that is $\boldsymbol{\pi}(t) = (\pi_n(t))_{n \in \mathbb{N}}$ where the subscript n indexes resulting sub-sticks. Then the stick-breaking process $\Pi = (\boldsymbol{\pi}(t))_{t \in \mathbb{R}^+}$ is an example of a (mass) fragmentation process. Motivated by this informal description, we define a fragmentation operator on sequences of real numbers in the general setting, and then give a formal definition of a (mass) fragmentation process over some space \mathcal{S} . We do this by adapting the formulation in [Bertoin, 2006, p. 119].

First consider the space \mathcal{S} of non-increasing non-negative sequences that sum to one given by $\mathcal{S} := \{\boldsymbol{\pi} = (\pi_i)_{i \in \mathbb{N}} \mid \pi_1 \geq \pi_2 \geq \dots \geq 0, \sum_{i \in \mathbb{N}} \pi_i = 1\}$. For each bounded sequence $(\pi_i)_{i \in \mathbb{N}}$ of non-negative real numbers we denote by $(\pi_i)_{i \in \mathbb{N}}^\downarrow$ the re-ordering of $(\pi_i)_{i \in \mathbb{N}}$ in a decreasing manner; we thus have that $(\pi_i)_{i \in \mathbb{N}}^\downarrow \in \mathcal{S}$ if and only if $\sum_{i \in \mathbb{N}} \pi_i = 1$. We now define a fragmentation operator on the space \mathcal{S} , and then give the definition of a fragmentation process (FP).

Definition 1.1 (Random Fragmentation Operator). Let $\text{Frag}(\cdot, \cdot)$ be a fragmentation operator defined as follows:

$$\text{Frag}(\boldsymbol{\pi}, (\bar{\boldsymbol{\pi}}^{(i)})_{i \in \mathbb{N}}) := \left(\pi_i \cdot \bar{\pi}_k^{(i)} \right)_{i,k \in \mathbb{N}}^\downarrow \quad (1.1)$$

where $(\bar{\boldsymbol{\pi}}^{(i)})_{i \in \mathbb{N}}$ are i.i.d. copies of some random sequence $\bar{\boldsymbol{\pi}}$. That is, for every integer i , $\text{Frag}(\cdot, \cdot)$ defines the distribution over the partitions of the i -th block π_i of $\boldsymbol{\pi}$ induced by the i -th i.i.d. copy $\bar{\boldsymbol{\pi}}^{(i)}$. The resulting partitions are the scaled sequences $\pi_i \cdot (\bar{\pi}_1^{(i)}, \bar{\pi}_2^{(i)}, \dots)$. Collecting these partitions for each π_i and rearranging them in decreasing order, we get the right hand side of Equation (1.1).

Definition 1.2 (Random Fragmentation Process, FP). We call an \mathcal{S} -valued Markov process $\boldsymbol{\pi}(t) := (\pi_n(t))_{n \in \mathbb{N}}$, a (mass) *fragmentation process* if the following two conditions hold:

- i. $\boldsymbol{\pi}(0) = (1, 0, 0, \dots)$.
- ii. For any $t, u \in \mathbb{R}^+$, conditioned on $\boldsymbol{\pi}(t)$, the random variable $\boldsymbol{\pi}(t+u)$ has the following distribution:

$$\boldsymbol{\pi}(t+u) \stackrel{d}{=} \text{Frag}(\boldsymbol{\pi}(t), (\boldsymbol{\pi}^{(i)}(u))_{i \in \mathbb{N}}) \quad (1.2)$$

where $\stackrel{d}{=}$ means equality in distribution.

In the fragmentation process, each sequence $\boldsymbol{\pi}(t)$ corresponds to a specific sorted split of a stick as brought in the stick-breaking example before. Intuitively, a fragmentation process can be understood through the stick-breaking example; in each splitting event the stick π_i is replaced with a (possibly infinite) sequence of shorter sticks that sum to π_i . The splitting event is independent of the splitting time, which in a more general setting would be given by a deterministic function. We will assume all sticks split concurrently according to such a function. The selection of the deterministic function used for the splitting rate, or the *divergence function*, will be explained further in the following Section. Note for practical purposes, in this work we focus on the *discrete* time FP, that is, splitting events are only allowed at discrete time steps (which corresponds to a fragmentation chain, c.f. Bertoin [2006]).

1.2 Nested Chinese Restaurant Processes

The nested Chinese restaurant processes [nCRP; Blei et al., 2010] is a Dirichlet “path-reinforcing” traverse of a tree where each data point starts at the root and descends to the leaves. More specifically, the first data point descends from the root and creates a new node with probability 1; the same data point repeats this process up to a pre-defined depth resulting in a leaf node (obtaining a chain graph). A later data point i starts from the root, and descends according to a Chinese restaurant processes (until maximum depth L). That is, if the data point reaches ω , it will either descend to an existing child or create a new child with probabilities:

$$p(\omega\omega_i|\omega) = \begin{cases} n_{\omega\omega_i}/(n_{\omega\cdot} + \alpha(|\omega|)) & \text{descends to child } \omega\omega_i \\ \alpha(|\omega|)/(n_{\omega\cdot} + \alpha(|\omega|)) & \text{creates a new child} \end{cases} \quad (1.3)$$

Here $n_{\omega\omega_i}$ denotes the number of data points descending from node ω to node $\omega\omega_i$ for all data points preceding data point i , and $n_{\omega\cdot}$ denotes a marginal count. This formulation leads to the well known “rich get richer” self-reinforcing property, which has been proved useful in various applications such as topic modelling and genetic mutation clustering [Teh, 2010].

Probability of the Combinatorial Structure For each node ω we refer to the set of ancestor nodes ($\omega': \Delta \preceq \omega' \preceq \omega$) – including the root and ω itself – as a path. The probability of each path is simply the product of probabilities given in Equation (??):

$$p(\Delta \rightarrow \omega) = \prod_{(\omega'\omega_i: \omega'\omega_i \preceq \omega)} p(\omega'\omega_i|\omega') \quad (1.4)$$

For each node, we refer to the collection of child nodes $\omega\omega_i$, and the counts associated with each child node as its branching structure. Since the branching structure is created by a CRP, we can write down the probability of the combinatorial branching structure analytically

$$g_{\omega} = p(n_{\omega\omega_i} : \forall i \mid n_{\omega\cdot}, \alpha) = \frac{\Gamma(\alpha)\alpha^{K_{\omega}} \prod_{\omega\omega_i} \Gamma(n_{\omega\omega_i})}{\Gamma(n_{\omega\cdot} + \alpha)} \quad (1.5)$$

where K_{ω} is the number of children nodes of ω , and α is the concentration parameter.

2 Dirichlet Fragmentation Processes

There exist many distributions satisfying the second condition set in Definition 1.2, each leading to a distinct family of fragmentation processes with different properties. One notable example of such distributions is the Poisson–Dirichlet (PD) distribution and its 2 parameter extension¹ [Pitman and Yor, 1997]. The PD distribution and its extensions have been shown to be powerful Bayesian nonparametric tools for mixture models (e.g. the popular Dirichlet process (DP) mixtures). Motivated by this success of the PD distribution, in this paper we derive a *Dirichlet fragmentation process* (DFP) defined as follows.

Definition 2.1 (DFP). We call a fragmentation process a *Dirichlet fragmentation process* if at each time t the Frag operator induces a Poisson–Dirichlet distribution over the partitions.

¹The 2-parameter PD distribution is also known as the Pitman–Yor process (PYP).

A useful property of the random fragmentation process is that it satisfies the Markov property – given a stick ω , subsequent fragmentation events are independent from ω ’s ancestors in the tree.

2.1 Recursive Stick-Breaking Construction

We gave an imprecise description of the stick breaking process in Section 1.1; now we give a formal definition to the process and use it as a constructive procedure for sampling from the Dirichlet fragmentation process. The stick-breaking process defined by Sethuraman [1994] is a constructive way for drawing samples from the DP. A random probability measure G can be drawn from a DP given a base probability measure H and concentration parameter α using a sequence of beta draws:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}, \quad \pi_k = \nu_k \prod_{i=1}^{k-1} (1 - \nu_i), \quad (2.1)$$

$$\nu_k \stackrel{\text{iid}}{\sim} \text{Beta}(1, \alpha), \quad \phi_k \sim H.$$

This can be viewed as taking a stick of unit length and breaking it at a random location. We call the left side of the stick π_1 and then break the right side at a new place, call the left side of this new break π_2 . We then continue this process of “keep the left piece and break the right piece again”. Sethuraman [1994] showed that the sequence of weights obtained from the stick breaking process (π_1, π_2, \dots) distributes according to the Poisson-Dirichlet distribution [Pitman and Yor, 1997]. Thus the stick breaking procedure can be used as a Dirichlet Frag operator. This is a useful property since we can apply this stick breaking Frag operator in a recursive way to induce a tree structure. This property has been noted and studied by Adams et al. [2010]. Here we provide a modified tree structured stick breaking procedure and use it as a way for sampling from the DFP.

Now we describe the recursive stick breaking process. The first step is to sample a beta random variable $\nu_{\omega} \sim \text{Beta}(1, \alpha)$ for each node in the tree with the exception of the root node. Then the length of the stick associated with node $\omega\omega_i$ is given by

$$\pi_{\omega\omega_i} = \pi_{\omega} \nu_{\omega\omega_i} \prod_{k=1}^{\omega_i-1} (1 - \nu_{\omega k}), \quad (2.2)$$

where π_{ω} is the stick length of the parent node. Through multiplying over beta variables of all prefixes of ω , the recursive definition given in Equation (??) can be unpacked as

$$\pi_{\omega} = \prod_{\omega' \omega_i \preceq \omega} \nu_{\omega' \omega_i} \prod_{k=1}^{\omega_i-1} (1 - \nu_{\omega' k}). \quad (2.3)$$

More generally, the concentration parameter α is allowed to vary for different nodes. For example, α can be a function of the depth of a given node, denoted by $\alpha(|\omega|)$. When the concentration parameter is infinitesimal for each node (e.g., $\alpha(|\omega|) = a(t_{\omega})dt$, whereas t can be seen as a fictitious time associated node ω), and the maximum depth of tree is sufficiently large, the recursive stick breaking will generate binary trees with probability 1. This special case of the DFP is known as the Dirichlet diffusion tree Neal [DDT, 2003]. Following a convention first introduced by Neal [2003], we shall call this function $\alpha(|\omega|)$

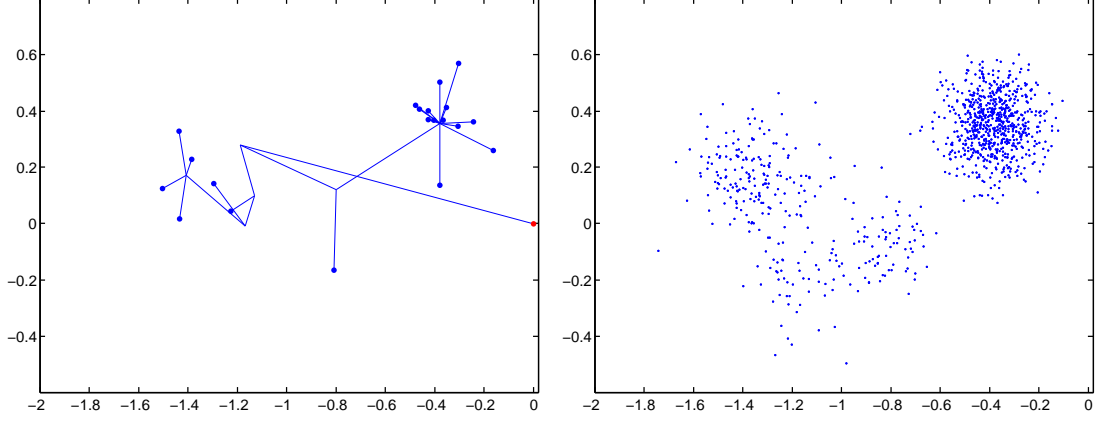


Figure 2: DFP Gaussian Diffusion Examples. Generation of a two-dimensional dataset from the Gaussian diffusion with the number of discrete time steps $L = 40$, $\sigma = 1$ and $\alpha(l)$ given by Equation (??) in the footnote. The plot on the left shows the first 20 data points generated, along with the underlying tree structure. The right plot shows 1000 data points obtained by continuing the procedure beyond these 20 data points.

the divergence function². The recursive stick-breaking process and the tree node indexing scheme are illustrated in Figure 1.

2.2 Parent-Child Transition Operators

Recall that for the Dirichlet process mixture model an unbounded number of partitions is generated where each partition is labelled with some parameter $\phi_k \sim H$. Given the generated data partition and corresponding labels, each data point is assumed to arise as a draw from a distribution $F(y|\phi_k)$, where ϕ_k is the k 'th component label from which y is generated. In the DFP we continue to assume that the data are generated independently given the latent labelling, but take advantage of the tree-structured partitioning of the data. That is, the distribution over the parameter at node $\omega\omega_i$, denoted $\phi_{\omega\omega_i}$, should depend on its parent ω . This parent-child dependence will be captured through a *Transition Operator*, denoted $T(\phi_{\omega\omega_i} \leftarrow \phi_\omega) := p(\phi_{\omega\omega_i}|\phi_\omega)$. For example, the Gaussian transition operator is given by

$$T(\phi_{\omega\omega_i} \leftarrow \phi_\omega) = \mathcal{N}(\phi_\omega, \sigma^2), \quad p(\Delta) = \mathcal{N}(0, \sigma^2) \quad (2.5)$$

where $p(\Delta)$ denotes the parameter distribution of the root node. An example of 1000 data points sampled from the DFP with a Gaussian transition operator is given in Figure 2.

3 A DFP Mixture Model

Given a DFP prior over the tree structure, we can obtain a hierarchical infinite mixture model by coupling the model with a mixture model component likelihood function, for

²An example of such a function is:

$$\alpha(l) = a((l+1)/L) - a(l/L), \quad (2.4)$$

where L is the number of discrete time steps, and a is defined as: $a(s) = \int_0^s c/(1-s)ds$, for some hyperparameter c .

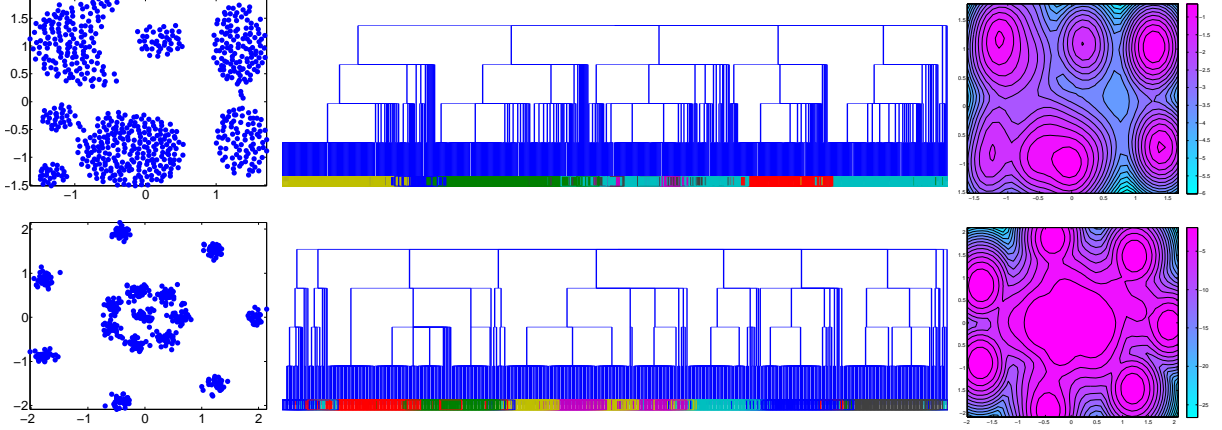


Figure 3: Results on Aggregation (top row), and R15 datasets (bottom row). The left plot shows the original data; the middle plot gives trees sampled from the posterior conditioning on the training data; the right plot shows the predictive densities resulting from our DFP mixture model.

example a Gaussian data distribution i.e.,

$$F(y|\phi_{\omega}) = \mathcal{N}(\phi_{\omega}, \sigma^2). \quad (3.1)$$

Here the subscript ω denote the index of the leaf node associated with data point y . We assume the dimensionality of the data to be 1 to keep notation simple. We will use this notation in the remaining part of the paper since the extension to arbitrary dimensionality is straightforward.

4 Inference by Gibbs Sampling

Recall our variables of interest; the variables y_i are our observations, and we let z_i denote the node (i.e. mixture component) from which y_i was generated – each y_i is assumed to arise as a draw of $F(y_i|\phi_{z_i})$. Here the vector $\phi = (\phi_{\omega})$ stores the parameters of each node. We use n_{ω} to denote the number of leaves descended from node ω , and K_{ω} to denote the number of children of node ω . Furthermore, we use ωk to denote the k 'th child of ω .

Let $\mathbf{y} = y_{1:N}$ be the sequence of data items, $\mathbf{y}_{\omega} = (y_i: z_i = \omega)$ be the sequence of data items generated from node ω , and $\mathbf{z} = z_{1:N}$ be the sequence of nodes generating \mathbf{y} . We attach a superscript to a set of variables or a count (e.g. \mathbf{y}^{-i} or n_{ω}^{-i}) to denote the removal of the variable corresponding to the superscripted index from the variable set or from the calculation of the count. In our examples $\mathbf{y}^{-i} := \mathbf{y} \setminus y_i$ and n_{ω}^{-i} is the number of observations (i.e. leaves) ultimately reached by node ω , leaving out data point y_i .

In the case of the Gaussian observation model, which is conjugate to the distribution of the leaf parameters, we integrate out the leaf and internal parameters ϕ in the sampling schemes. Denote the conditional density of y_i under leaf node z given all data points except y_i as $f_z^{-y_i}(y_i)$. The non-conjugate case can be tackled by adapting similar techniques to the ones developed for non-conjugate DP mixtures [Neal, 2000].

Finally we specify priors on the hyper-parameters of the divergence function (Equation (??) in the footnote), c , and the diffusion precision τ (the inverse of σ^2 in Equation (??)):

$$c \sim \text{Gamma}(a_c, b_c), \quad \tau \sim \text{Gamma}(a_{\tau}, b_{\tau}) \quad (4.1)$$

Here $\text{Gamma}(a, b)$ is a Gamma distribution with shape a and rate b . In all experiments we used $a_c = 1, b_c = 1, a_\tau = 1, b_\tau = 1$. Next we describe a Gibbs sampler for the DFP.

Step 1: Sampling \mathbf{z} . This can be realised by

$$p(z_i = \omega | \mathbf{z}^{-i}, c, \tau) \propto \begin{cases} p_\omega^{-i} f_\omega^{-y_i}(y_i) & \text{if } \omega \text{ is an existing leaf node} \\ p_{\omega'}^{-i} \frac{\alpha_{\omega'}}{\alpha_{\omega'} + n_{\omega'}} f_{\omega'}^{-y_i}(y_i) & \text{if } \omega \text{ is a new leaf node} \end{cases} \quad (4.2)$$

with ω' parent of ω .

Here p_ω^{-i} is the probability of reaching node ω from the root node leaving out y_i (as defined in Equation (??)), and α_ω is the divergence function evaluated at depth $|\omega|$. Intuitively, the above equation defines the two ways that y_i can be generated. In the first way, data item i follows an existing branch until it reaches a leaf node ω , which has probability p_ω^{-i} . Then this probability is multiplied with the likelihood term, giving us the total probability that y_i is generated from node ω . In the second way, data item i initially follows an existing branch until it reaches (internal) node ω' , then it diverges from the current branch and creates a new leaf node, for which the total probability is simply the product of the probability of reaching node ω' , and the probability of diverging from ω' . Lastly, multiplied with a likelihood term, this gives us the probability of y_i being generated from a new node. Note that updating the leaf assignment of each data point y_i will also update the count vector $n_{\omega\cdot}$, and vice versa. In fact, this is the only way that \mathbf{z} influences the other variables, i.e. ϕ and c .

Step 2: Sampling divergence function hyperparameter \mathbf{c} . The probability of the tree structure given the divergence function is simply the product of the probabilities of the branching structures for each internal node. Since at each internal node ω the process of descending to the children follows a CRP, the probability of a branching structure for each internal node g_ω is given by Equation (??). Coupled with the gamma prior, the Gibbs conditional probability for c is

$$p(c | \tau, \mathbf{g}, \mathbf{n}, \phi) \propto \text{Gamma}(a_c, b_c) \prod_{(\omega: \text{ all internal nodes})} g_\omega. \quad (4.3)$$

Step 3: Sampling the precision τ . It is straightforward to sample τ given the node parameters ϕ . The probability of all node parameters $p(\phi)$ is simply the product of a set of Gaussians, since each node's parameter distribution $p(\phi_{\omega\omega_i} | \phi_\omega)$ is Gaussian. Coupled with a gamma prior, the Gibbs conditional probability for the precision τ is given by

$$p(\tau | c, \mathbf{g}, \mathbf{n}, \phi) \propto \text{Gamma}(a_\tau, b_\tau) \times \prod_{(\omega: \text{ all internal nodes})} \prod_{(\omega\omega_i: \text{ children of } \omega)} \text{Gamma}\left(1, \frac{(\phi_{\omega\omega_i} - \phi_\omega)^2}{2}\right). \quad (4.4)$$

In summary, for each observation the proposed Gibbs sampler iteratively samples a path leading to it conditioned on paths leading to remaining observations (note this is different from the Gibbs sampler for the nCRP topic model, which samples path leading to each observation in two separate steps, see Blei et al. [2010] for details). Most existing inference procedures for trees employ a “prune-graft” algorithm; that is, first remove a subtree and then propose to re-attache the sub-tree elsewhere. The proposal is then

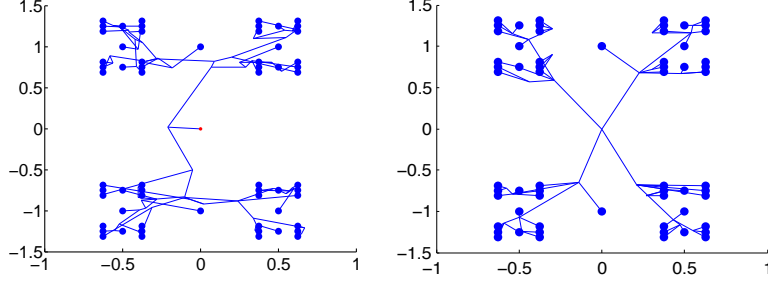


Figure 4: Hierarchical clustering results on the Synthetic dataset. Left: tree structures sampled from the DDT model conditioning on the data. Right: tree structures sampled from the DFP mixture model conditioning on the data.

DATASET	GMM	DPM	DDT	DFP mixture
R15	-2.127 ± 0.158	-0.759 ± 0.122	-0.861 ± 0.123	-0.705 ± 0.086
D31	-2.593 ± 0.036	-1.790 ± 0.040	-1.798 ± 0.044	-1.654 ± 0.022
AGGR.	-2.151 ± 0.076	-2.064 ± 0.063	-2.091 ± 0.057	-1.431 ± 0.008
MACA.	-15.039 ± 0.584	-15.145 ± 0.611	-14.816 ± 0.546	-12.725 ± 0.127
CCLE	-4.8183 ± 0.947	-4.6036 ± 0.331	-3.7266 ± 0.457	-2.825 ± 0.495

Table 1: Predictive log likelihood (\log_e) for GMM, DP mixture, DDT, and DFP mixture.

accepted or rejected using an MH step. As we will show in the following section, empirically this Gibbs sampler results in significantly improved performance when compared to state-of-the-art models using this “prune-graft” inference for both hierarchical clustering and density estimation.

5 Experiments

In this section we describe two sets of experiments to highlight the two aspects of the discrete time DFP mixture model: its hierarchical nature and its nonparametric density modelling nature. To demonstrate the hierarchical nature of the DFP we compare the model to the agglomerative clustering algorithm. For the DFP, we implemented the inference algorithm described in Section 4. The software implements the discrete DFP with arbitrary depth, and is available at [URL]. We use Neal’s Flexible Bayesian Modelling (FBM) package for the DDT and Matlab’s implementation for the agglomerative clustering algorithm.

5.1 Hierarchical Clustering

First we compare the DFP mixture model to the agglomerative clustering algorithm. We performed experiments on four datasets (one hand crafted synthetic dataset and three real datasets). The real datasets we used are R15 (600 examples, 2 attributes [Veenman et al. \[2002\]](#)), Aggregation (referred to as AGGR, 788 examples, 2 attributes, [Veenman et al. \[2002\]](#)), and Glass (214 examples, 7 classes, 9 attributes). For the synthetic dataset, trees sampled from the posterior of the DFP mixture model and the DDT conditioning on the training data are shown in Figure 4. Both methods find a good hierarchical clustering of the data items. While the DDT is forced to choose a binary branching structure over the clusters, the DFP can represent a more parsimonious solution. Such parsimonious solutions are more interpretable and potentially lead to better explanations for the data.

Similar results are also observed for the real datasets. The results on the AGGR and R15 datasets are shown in Figure 3. As we can see from Figure 3, most data points with the same class label are merged in the first level of the DFP mixture model, which leads to a clean summary of the structure of the data.

Furthermore, in order to assess the quality of these hierarchical clustering results, we also computed the tree purity score for various algorithms on the Glass dataset; the tree purity score was introduced by Heller and Ghahramani [2005] and motivated as a reasonable metric for evaluating hierarchical clustering algorithms. On the Glass dataset the purity scores are 0.5064 (DFP), 0.4815 (agglomerative, average linkage), and 0.4568 (DDT). The result of the agglomerative algorithm are consistent with those reported in Heller and Ghahramani [2005]. However, while Heller and Ghahramani [2005]’s Bayesian Hierarchical Clustering algorithm exhibits lower purity score when compared to the agglomerative algorithm on the Glass dataset, the DFP mixture model produces a slightly better one.

5.2 Density Estimation

To evaluate the power of the DFP in density estimation, we compare the DFP mixture model to traditional mixture models including the Gaussian Mixture Model (GMM), the Dirichlet Process Mixtures (DPM), and the Dirichlet Diffusion Tree (DDT) over 5 datasets. The 5 datasets we used are the macaque skull measurements (MACA, 228 examples, 10 attributes), R15, Aggregation, D31 (3100 examples, 2 attributes), and the Cancer cell line encyclopedia (CCLE, 504 examples, 24 attributes). In particular, the CCLE dataset consists of measurements of the sensitivity of 504 cancer derived cell lines to 24 drugs. Such data has the potential to help biologists understand the relationship between different cancer types and drug effects, and to aid in clinical practice [Barretina et al., 2012].

For all datasets we train each model using 90% of the data and report the predictive log likelihood for the remaining 10% of the data. For the DFP, we set the depth of the tree at $L = 4$. For all methods under comparison, we run the MCMC inference algorithm until the predictive log likelihood for the train data converges. As shown in Table 1, on all datasets the DFP mixture model obtains the highest predictive log likelihood. For the MACA dataset, the DFP mixture model outperforms all previous models: the performance of the model is 2.5 orders better (on \log_e scale). This is a significant improvement as previous attempts on the same dataset only obtained a small improvement, as reported in Knowles and Ghahramani [2011] and Adams et al. [2008]. The improvement of the DFP over existing methods is consistent with all other datasets we tried, in particular, the performance on the CCLE is about 1 order better.

6 Discussion

This paper have presented the Dirichlet fragmentation process for modelling tree structures. The DFP is derived as a useful variant of fragmentation processes, and is connected to a number of existing models such as Neal [DDT 2003], Blei et al. [nCRP 2010], Adams et al. [TSSB 2010], Knowles and Ghahramani [PYDT 2011], Rodriguez et al. [nDP 2008]. Particularly, we derived a simple hierarchical mixture model based on the DFP, and an efficient Gibbs-style sampler. This DFP hierarchical mixture model generalises the popular Dirichlet process mixture model. Unlike the latter, which partitions data into a flat layer of clusters, the DFP mixture model organises clusters into a tree structure. Not only this

provides more interpretable summary of the data, but also leads to significantly better accuracy as demonstrated in the density estimation experiments. Future theoretical work will study the connection between the DFP and hierarchical DPs, and extends the DFP to model group data and sequential data.

References

- Jean Bertoin. *Random fragmentation and coagulation processes*, volume 102 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2006. ISBN 978-0-521-86728-3; 0-521-86728-2. doi: 10.1017/CBO9780511617768.
- David Aldous. The continuum random tree. i. *The Annals of Probability*, pages 1–28, 1991.
- Peter McCullagh, Jim Pitman, and Matthias Winkel. Gibbs fragmentation trees. *Bernoulli*, pages 988–1002, 2008.
- Yee W Teh, Charles Blundell, and Lloyd Elliott. Modelling genetic variations using fragmentation-coagulation processes. In *Advances in neural information processing systems*, pages 819–827, 2011.
- David M Blei, Thomas L Griffiths, and Michael I Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of the ACM (JACM)*, 57(2):7, 2010.
- Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010.
- Jim Pitman and Marc Yor. The two-parameter Poisson–Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.
- J. Sethuraman. A constructive definition of Dirichlet priors. *Statist. Sinica*, 4:639–650, 1994.
- Ryan Prescott Adams, Zoubin Ghahramani, and Michael I Jordan. Tree-structured stick breaking for hierarchical data. *Advances in Neural Information Processing Systems*, 23: 19–27, 2010.
- Radford M Neal. Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics*, 7:619–629, 2003.
- Radford M Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.
- Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. A maximum variance cluster algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9): 1273–1280, 2002.
- Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.
- Jordi Barretina, Giordano Caponigro, Nicolas Stransky, Kavitha Venkatesan, Adam A Margolin, Sungjoon Kim, Christopher J Wilson, Joseph Lehár, Gregory V Kryukov, Dmitriy Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

- David A Knowles and Zoubin Ghahramani. Pitman–Yor diffusion trees. *27nd Conference on Uncertainty in Artificial Intelligence.*, 2011.
- Ryan P Adams, Iain Murray, and David MacKay. The Gaussian process density sampler. In *Advances in Neural Information Processing Systems*, pages 9–16, 2008.
- Abel Rodriguez, David B Dunson, and Alan E Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103(483), 2008.